What is a **dbt Snapshot**?

A **dbt snapshot** is a way to **track changes in records over time** — especially useful when dealing with **slowly changing dimensions (SCD)** or **historical changes** to rows in a source table in your source data.

**Snapshots help by:**

**Preserving history** (SCD Type 2 style) – Keep tracks of all History of changes.

Allowing you to **see how records evolve**

Supporting **auditing, compliance, and reporting needs**

**How It Works (At a High Level) :**

* You define a snapshot using a SQL query and config.
* dbt runs that query, compares new results to old ones.
* If changes are found (based on your strategy), it inserts a new version of the row.
* Stores metadata columns like dbt\_valid\_from and dbt\_valid\_to.

**Detailed Steps for Implementing or Adding Snapshots:**

**1. Create a Snapshot Directory**

If not already present:

mkdir snapshots

Inside your dbt\_project.yml, make sure you’ve included this path:

snapshot-paths: ["snapshots"]

2. **Create a Snapshot File**

Create a file like snapshots/customers\_snapshot.sql:

{% snapshot customers\_snapshot %}

{{

config(

target\_schema='snapshots',

unique\_key='customer\_id',

strategy='check',

check\_cols=['XXX', 'YYYYYY'],

)

}}

select \* from {{ source('SRC1', 'MODEL1') }}

{% endsnapshot %}

**Explanation of Config:**

* target\_schema: Where the snapshot table will be created.
* unique\_key: Unique row identifier.
* strategy: check compares values in check\_cols, timestamp compares based on a timestamp column.
* check\_cols: List of columns to watch for changes.

3. **Run the Snapshot**

**Command for Running Snapshots:**

dbt snapshot

This will:

Create the snapshot table if not already present.

Compare the current source data to the existing snapshot.

Insert new records if changes are detected.

### 4. ****Querying the Snapshot Table****

The snapshot table will contain:

* dbt\_valid\_from: When the record became valid.
* dbt\_valid\_to: When the record was no longer valid (null if current).
* All the original columns from your source.

select \*

from {{ ref('customers\_snapshot') }}

where dbt\_valid\_to is null -- get latest records

Note:

* Don’t forget to commit your new snapshot file to git.
* Add tests for uniqueness on customer\_id in your snapshot model.
* Schedule dbt snapshot to run regularly in pipeline